NASA/CR-2004-213245
NIA Report No. 2004-08

State-Dependent Riccati Equation Regulation of Systems with State and Control Nonlinearities

Scott C. Beeler
National Institute of Aerospace, Hampton, Virginia

July 2004

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- Email your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2004-213245
NIA Report No. 2004-08

State-Dependent Riccati Equation Regulation of Systems with State and Control Nonlinearities

Scott C. Beeler
National Institute of Aerospace, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NCC-1-02043

July 2004

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

STATE-DEPENDENT RICCATI EQUATION REGULATION OF SYSTEMS WITH STATE AND CONTROL NONLINEARITIES

Scott C. Beeler*

ABSTRACT

The state-dependent Riccati equation (SDRE) is the basis of a technique for suboptimal feedback control of a nonlinear quadratic regulator (NQR) problem. It is an extension of the Riccati equation used for feedback control of linear problems, with the addition of nonlinearities in the state dynamics of the system resulting in a state-dependent gain matrix as the solution of the equation. In this paper several variations on the SDRE-based method will be considered for the feedback control problem with control nonlinearities. The control nonlinearities may result in complications in the numerical implementation of the control, which the different versions of the SDRE method must try to overcome. The control methods will be applied to three test problems and their resulting performance analyzed.

1 INTRODUCTION

The nonlinear quadratic regulator (NQR) problem can be approached in many different ways, using different approximation techniques or numerical schemes to obtain a feedback control as near to optimal as possible. A direct formula for an optimal feedback control is almost always impossible for nonlinear systems, so several suboptimal methods have been proposed in the literature. Among these is the method involving the state-dependent Riccati equation (SDRE), based on the constant-valued Riccati equation used to find the optimal feedback control for linear systems. The SDRE extends this approach to the nonlinear case by allowing the matrices involved to be functions of the state variables, and possibly the controls as well. The SDRE method has the advantage of being fairly simple numerically in comparison to many other techniques, as well as being closely related to the well-understood Riccati equation for linear problems.

Early work on the state-dependent Riccati equation was done by Pearson [1], Garrard, McClamroch and Clark [2], Burghart [3], and Wernli and Cook [4]. More recently, the SDRE control has been studied by Krikelis and Kiriakidis [5], and Cloutier, D'Souza and Mracek [6, 7] (with the SDRE applied to a nonlinear benchmark problem in [8]). Hammett, Hall and Ridgely look at controllability issues for the SDRE in [9]. A comparison study involving the SDRE among other nonlinear control methods was done in [10], and tracking control and state estimation methods using the SDRE were developed in [11].

Most of these methods assume a system in a control-affine form such as

$$\dot{x} = f(x) + B(x)u, \tag{1}$$

*Staff Scientist, National Institute of Aerospace (NIA), Hampton, VA 23666. Email: scbeeler@nianet.org.

where there are state nonlinearities but only linear dependence on the control inputs. Of the above cited papers, only Wernli and Cook [4] consider the more general case

$$\dot{x} = f(x, u). \tag{2}$$

This type of system leads to significant complications in the SDRE method control equations, making it much more difficult to find a direct formula for u in terms of x. However, since there are many applications where the control inputs are nonlinear, we wish to extend the state-dependent Riccati equation method to be applicable to the wider case of equation (2), and so in this paper we will be studying various techniques to allow control of this more complex problem.

The approach of Wernli and Cook is to split the state and control dynamics matrices into constant and variable parts, and then use those to solve for a truncated power series approximation to the gain matrix solution of the SDRE. This allows the numerical calculations to be comparatively simple and mostly done offline prior to the application of the control to the system. This approach was also used for problems with nonlinear state dynamics in [10], and will be one of the techniques used in this paper. An alternate approach is to do most of the calculations online, solving the SDRE repeatedly to update the control as the state changes. This has the advantage of solving the exact equation rather than an approximation, but it must do so constantly during the control implementation rather than a single time prior to the start of the problem. One way of working around control nonlinearities is to move them into the state dynamics by reformulating the system with a cheap control (or integral control) approach, as suggested in [12]. There are some numerical complications to these techniques which may favor one over the others in certain circumstances, as will be discussed later in this paper.

A summary of the derivation and nature of the state-dependent Riccati equation and its use for nonlinear feedback control problems will be given in Section 2. The two primary numerical realizations of the nonlinear-u SDRE to be considered here will then be described: the power series approach in Section 3 and the online control update approach in Section 4. Several versions of these control formulations will be applied to three example problems with the results discussed in Sections 5-7. An evaluation of the SDRE techniques and overall conclusions will be given in Section 8.

2 BASIS OF THE STATE-DEPENDENT RICCATI EQUATION

The state-dependent Riccati equation (SDRE), as part of a suboptimal feedback control for nonlinear systems, has its roots in the optimality conditions for the NQR problem. This parallels the constant-valued algebraic Riccati equation's status in finding the optimal feedback control for a given linear quadratic regulator (LQR) problem. The nonlinear system we will study is of the form

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & f(x(t)) + g(x(t), u(x(t))) \\ x(0) & = & x_0, \end{array} \right. \tag{3}$$

with $x \in R^m$ and $u : R^m \to R^k$. The objective in this problem is to find the feedback control $u(x)$ which minimizes the quadratic cost functional

$$J(x_0, u) = \frac{1}{2} \int_0^\infty \left(x^T Q x + u^T R u \right) dt \tag{4}$$

with given constant-valued state and control weight matrices Q (symmetric, positive semi-definite) and R (symmetric, positive definite).

The SDRE is derived from the optimality conditions on the Hamiltonian for this problem, which is defined as

$$\mathcal{H}(x,u,p) = \frac{1}{2}x^T Q x + \frac{1}{2}u^T R u + p^T \left(f(x) + g(x,u)\right). \tag{5}$$

From the Hamiltonian the necessary conditions for the optimal control of the dynamic system and cost functional in equations (3-4) are given in terms of x, u, and the costate $p \in R^m$ by

$$\begin{aligned} \dot{x} &= \frac{\partial \mathcal{H}}{\partial p} = f(x) + g(x,u) && (6)\\ \dot{p} &= -\frac{\partial \mathcal{H}}{\partial x} = -Qx - \frac{\partial f^T}{\partial x}(x)p - \frac{\partial g^T}{\partial x}(x,u)p && (7)\\ 0 &= \frac{\partial \mathcal{H}}{\partial u} = Ru + \frac{\partial g^T}{\partial u}(x,u)p. && (8) \end{aligned}$$

(See references [13, 14], for example.) From equation (8), the control can be written as

$$u(t) = -R^{-1}\frac{\partial g^T}{\partial u}(x(t),u(t))p(t). \tag{9}$$

In a linear problem, where $f(x) = A_0 x$ and $g(x,u) = B_0 u$, we would look for a costate solution of the form $p(t) = \Pi_0 x(t)$. We would take the time derivative of this costate formula, substitute in equations (6,7,9) and the chosen formula for p, and obtain a linear feedback control $u(x) = -R^{-1}B_0^T \Pi_0 x$, with the gain Π_0 given by the solution to the constant-valued Riccati equation

$$\Pi_0 A_0 + A_0^T \Pi_0 - \Pi_0 B_0 R^{-1} B_0^T \Pi_0 + Q = 0, \tag{10}$$

which is well understood and for which good solvers are available.

For the more general nonlinear case the SDRE method mimics the above approach by rewriting the functions in (3) in quasi-linear form as $f(x) = A(x)x$ and $g(x,u) = B(x,u)u$. (Note that the choices of the matrix functions A and B are not unique, and different sub-optimal SDRE controls will result from using different choices of A and B in the manner described below.) This "quasi-linearization" rewriting f and g has also been called an "apparent linearization", "extended linearization", and "state-dependent coefficient parameterization" in various places. With these new forms of f and g, we seek a costate of the form

$$p(t) = \Pi\left(x(t), u(t)\right) x(t). \tag{11}$$

With a substitution of the above p and g formulas into equation (9), the control equation becomes

$$u(x) = -R^{-1}B^T(x,u)\Pi(x,u)x - R^{-1}\sum_{i=1}^{k} u_i \left(\frac{\partial B_{1\to m,i}}{\partial u}(x,u)\right)^T \Pi\left(x,u\right)x. \tag{12}$$

The B-column derivative with respect to u,

$$\frac{\partial B_{1\to m,i}}{\partial u} = \begin{pmatrix} \partial B_{1i}/\partial u_1 & \cdots & \partial B_{1i}/\partial u_k \\ \vdots & \ddots & \vdots \\ \partial B_{mi}/\partial u_1 & \cdots & \partial B_{mi}/\partial u_k \end{pmatrix}, \tag{13}$$

will be considered small and discarded in finding the basic SDRE control, in order to simplify the calculations involved and keep the form of the control analogous to the constant-valued Riccati equation.

Taking the derivative of equation (11) yields

$$\dot{p} = \Pi\left(x,u\right)\dot{x} + \left[D_t \Pi\left(x,u\right)\right]x, \tag{14}$$

where the term $D_t\Pi(x,u)$ represents the total time derivative of $\Pi(x(t),u(x(t)))$ given by

$$D_t\Pi(x,u) = \sum_{i=1}^{m} \frac{\partial \Pi}{\partial x_i}(x,u)\dot{x}_i + \sum_{i=1}^{k} \frac{\partial \Pi}{\partial u_i}(x,u)\dot{u}_i. \tag{15}$$

We then substitute into equation (14) the formulas for $\dot{x}$, $\dot{p}$, u and p determined by equations (6,7,9,11), along with the quasi-linear formulas for f and g given above. This results in

$$\begin{aligned}
\Pi\left(x,u\right)&\left[A(x)x - B(x,u)R^{-1}\left(B^T(x,u) + \sum_{i=1}^{k} u_i \left(\frac{\partial B_{1\to m,i}}{\partial u}(x,u)\right)^T\right)\Pi\left(x,u\right)x\right] \\
&+ \left[D_t\Pi\left(x,u\right)\right]x = -Qx - \left[A^T(x) + \textstyle\sum_{i=1}^{m} x_i \left(\frac{\partial A_{1\to m,i}}{\partial x}(x)\right)^T\right. \\
&\qquad\qquad \left. + \textstyle\sum_{i=1}^{k} u_i \left(\frac{\partial B_{1\to m,i}}{\partial x}(x,u)\right)^T\right]\Pi\left(x,u\right)x,
\end{aligned} \tag{16}$$

where the A- and B-column derivatives are given similarly to (13) by

$$\frac{\partial A_{1\to m,i}}{\partial x} = \begin{pmatrix} \partial A_{1i}/\partial x_1 & \cdots & \partial A_{1i}/\partial x_m \\ \vdots & \ddots & \vdots \\ \partial A_{mi}/\partial x_1 & \cdots & \partial A_{mi}/\partial x_m \end{pmatrix}, \tag{17}$$

$$\frac{\partial B_{1\to m,i}}{\partial x} = \begin{pmatrix} \partial B_{1i}/\partial x_1 & \cdots & \partial B_{1i}/\partial x_m \\ \vdots & \ddots & \vdots \\ \partial B_{mi}/\partial x_1 & \cdots & \partial B_{mi}/\partial x_m \end{pmatrix}. \tag{18}$$

By rewriting equation (16) and factoring out x, we obtain

$$\begin{aligned}
&\left[\Pi(x,u)A(x) + A^T(x)\Pi(x,u) - \Pi(x,u)B(x,u)R^{-1}B^T(x,u)\Pi(x,u) + Q\right] \\
&+ \left[\sum_{i=1}^{m} x_i \left(\frac{\partial A_{1\to m,i}}{\partial x}(x)\right)^T \Pi\left(x,u\right) + \sum_{i=1}^{k} u_i \left(\frac{\partial B_{1\to m,i}}{\partial x}(x,u)\right)^T \Pi\left(x,u\right)\right. \\
&\left. - \Pi\left(x,u\right)B(x,u)R^{-1}\sum_{i=1}^{k} u_i \left(\frac{\partial B_{1\to m,i}}{\partial u}(x,u)\right)^T \Pi\left(x,u\right) + D_t\Pi\left(x,u\right)\right] = 0.
\end{aligned} \tag{19}$$

The first part of this equation is the state-dependent Riccati equation, and the second part is a set of extra terms which, as in equation (12), will be assumed to be small and eliminated. These modifications mean that the SDRE will find only a suboptimal control. How significant the difference will be depends on the specific problem, with the greater divergence from the linearized form generally resulting in further-from-optimal controls. With the

elimination of the derivative terms, the SDRE control will be given by the following Riccati equation and control formula:

$$\Pi(x,u)A(x)+A^T(x)\Pi(x,u)-\Pi(x,u)B(x,u)R^{-1}B^T(x,u)\Pi(x,u)+Q=0, \tag{20}$$

$$u(x)=-R^{-1}B^T(x,u)\Pi(x,u)x. \tag{21}$$

3 POWER SERIES FORMULATION

The nature of the formulation in equations (20-21) creates some significant difficulties in finding a solution, specifically that the Riccati equation (20) is dependent on both the state and the control, and that the control equation (21) is not a direct expression for u either. A direct algebraic solution of these equations is impossible for virtually all cases, so we must find a good means of approximation or numerical solution of the equations.

One approach, from Wernli and Cook [4], is to take a power series expansion for $\Pi(x)$ in terms of a temporary variable ε:

$$\Pi(x,u,\varepsilon)=\sum_{j=0}^{\infty}\varepsilon^j L_j(x,u), \tag{22}$$

then split A and B into constant and variable parts:

$$A(x,\varepsilon)=A_0+\varepsilon\Delta A(x),\;\; B(x,u,\varepsilon)=B_0+\varepsilon\Delta B(x,u), \tag{23}$$

and substitute these into the state-dependent Riccati equation (20). Proceed to separate out by powers of ε and solve the resulting series of equations for as many L_j terms as desired:

$$\begin{aligned}
L_0A_0+A_0^TL_0-L_0B_0R^{-1}B_0^TL_0+Q &= 0 &(24)\\
L_1\left(A_0-B_0R^{-1}B_0^TL_0\right)+\left(A_0^T-L_0B_0R^{-1}B_0^T\right)L_1+L_0\Delta A+\Delta A^TL_0 & \\
-L_0\left(B_0R^{-1}\Delta B^T+\Delta BR^{-1}B_0^T\right)L_0 &= 0 &(25)\\
L_j\left(A_0-B_0R^{-1}B_0^TL_0\right)+\left(A_0^T-L_0B_0R^{-1}B_0^T\right)L_j+L_{j-1}\Delta A+\Delta A^TL_{j-1} & \\
-\sum_{i=1}^{j-1}\left(L_iB_0R^{-1}B_0^TL_{j-i}\right)-\sum_{i=0}^{j-1}L_i\left(B_0R^{-1}\Delta B^T+\Delta BR^{-1}B_0^T\right)L_{j-1-i} & \\
-\sum_{i=0}^{j-2}L_i\Delta BR^{-1}\Delta B^TL_{j-2-i} &= 0. &(26)
\end{aligned}$$

These L_j matrices are then substituted into the control equation (with ε set to 1), which for N_p nonlinear terms becomes

$$u(x)=-R^{-1}B^T(x,u)\sum_{j=0}^{N_p}L_j(x,u)x. \tag{27}$$

Power series approaches are also described in [2, 3, 5], but not for systems containing control nonlinearities.

Note that increasing the number of power series terms used will not necessarily improve the performance of the control algorithm. For one thing, even an exact solution of the SDRE

produces only a suboptimal control, so a closer approximation to the solution may not result in a closer-to-optimal control. For another, the performance will vary as the distance from the expansion point varies; in particular, farther from this point the higher order terms will likely become less accurate and end up detracting from the effectiveness of the control.

In most cases with only state (not control) nonlinearities the computations in the power series expansion are relatively easy to do, since in that situation B, Π, and L_j are dependent only on x. This allows all the major numerical work to be performed offline, leaving just the direct evaluation of the $L_j(x)$ functions at the current state value to be done during the control implementation on the subject system. With nonlinear control inputs, however, this approach is more difficult. For simple functions of u (such as $B(x,u) = B_1(x) + B_2(x)u$) and a small number of power series terms, it is possible to rewrite equation (27) as a low-order polynomial in u and solve this online for the control value at the current x. Choosing which root of the polynomial to use is nontrivial, however, and in this implementation it has been done both by using the root with the smallest absolute value (to minimize excessive control use) and by using the root closest to that in the previous time step (to maintain the continuity of the control inputs as much as possible). Alternatively equation (27) can be solved iteratively, also online, for u:

$$u_{(n+1)}(x) = -R^{-1}B^T(x, u_{(n)}) \sum_{j=0}^{N_p} L_j(x, u_{(n)})x. \tag{28}$$

The starting point $u_{(0)}$ for this iteration process would be a chosen value such as u at the previous time step, or $u = L_0 x$, or $u = 0$.

Another approach is to change the structure of the problem by using a cheap control formulation, shifting the control variables in the original problem into additional state variables, and using the derivatives of the original controls as the new controls. This approach is also proposed in [12]. The new system is written as:

$$\begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} A(x) & B(x,u) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ I_k \end{bmatrix} \tilde{u}, \tag{29}$$

where I_k is a $k \times k$ identity matrix. The cost functional is rewritten as

$$J(x_0, \tilde{u}) = \frac{1}{2} \int_0^\infty \left(\begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \tilde{u}^T \tilde{R} \tilde{u} \right) dt. \tag{30}$$

The weight $\tilde{R}$ on the new control vector $\tilde{u}$ is set very close to zero to keep this new cost functional close to the original, but must also not be set too small or else computational problems will result. Setting $\tilde{R} = 0$ exactly would in theory keep the system dynamics the same as the original, but this would make the implementation of feedback control formulas such as equations (20-21) impossible, as they would then contain an undefined $\tilde{R}^{-1}$. Extremely small but nonzero $\tilde{R}^{-1}$ values will create problems as well. The cheap control approach makes the control problem computationally simpler by combining all the nonlinearities into the state dynamics and leaving the new control inputs linear. With the new $\tilde{B} = [0\ I_k]^T$ not dependent on $\tilde{u}$, the calculation of the control using the L_j matrices can be done without the need for a polynomial solution or iterative process.

The power series method has the advantage that most of its calculations are done offline, with only the polynomial solution or iteration for u (both of which are fast) performed during the control implementation. However, the control may be less optimal than desired, since the SDRE generates a suboptimal form of the control to begin with, and the use of a power series results in a further approximation. The best number of power series terms to use is also difficult to determine, and may vary from application to application.

4 ONLINE CONTROL UPDATE FORMULATION

As an alternative to the power series approach discussed in the previous section, we can use a type of method which does most of the numerical calculations online, during the control implementation. The control equations (20-21) can be solved repeatedly as the system evolves to update the gain Π and control u:

$$\Pi_{n+1}A(x_n) + A^T(x_n)\Pi_{n+1} - \Pi_{n+1}B(x_n,u_n)R^{-1}B^T(x_n,u_n)\Pi_{n+1} + Q = 0, \tag{31}$$

$$u_{n+1} = -R^{-1}B^T(x_n,u_n)\Pi_{n+1}x_n. \tag{32}$$

(Here x_n and u_n give the state and control values at time step n.) These evaluations can be done a single time, or, in an effort to obtain a more accurate solution, in an iterative loop back and forth between the two equations given by

$$\Pi_{n+1,k}A(x_n) + A^T(x_n)\Pi_{n+1,k} - \Pi_{n+1,k}B(x_n,u_{n+1,k})R^{-1}B^T(x_n,u_{n+1,k})\Pi_{n+1,k} + Q = 0, \tag{33}$$

$$u_{n+1,k+1} = -R^{-1}B^T(x_n,u_{n+1,k})\Pi_{n+1,k}x_n, \tag{34}$$

starting from an initial iterate such as $u_{n+1,0} = u_n$. (Here n is the time step number and k is the iteration number at the current time step.)

In implementation on a physical system, where the control calculations must be done in real time, the gain and control updates must be done only at some discrete intervals to allow time for the computations to be done. This will produce a less effective control than with constant updates, but if the control can be reevaluated quickly enough this approach may still result in better performance than the power series method, since it solves the Riccati equation exactly, while the power series finds an approximate solution. For investigating the performance of the control method on a simulated problem, however, the updates can be done at every time step in the ODE solution (as will be done on the test examples in the sections to follow).

While deriving the power series approximation in equations (24-27), the extra derivative terms of A, B and Π are dropped from equations (19-12) so that the control formula remains in Riccati equation form and is more easily solveable. However, when updating the control online, as in equations (31-32) or (33-34), as a variation on the control method we can retain the extra derivative terms and instead solve

$$\begin{aligned}
&\Pi_{n+1}A(x_n) + A^T(x_n)\Pi_{n+1} - \Pi_{n+1}B(x_n,u_n)R^{-1}B^T(x_n,u_n)\Pi_{n+1} + Q \\
&+\sum_{i=1}^{m}(x_n)_i\left(\frac{\partial A_{1\to m,i}}{\partial x}(x_n)\right)^T\Pi_{n+1} + \sum_{i=1}^{k}(u_n)_i\left(\frac{\partial B_{1\to m,i}}{\partial x}(x_n,u_n)\right)^T\Pi_{n+1} \\
&-\Pi_{n+1}B(x_n,u_n)R^{-1}\sum_{i=1}^{k}(u_n)_i\left(\frac{\partial B_{1\to m,i}}{\partial u}(x_n,u_n)\right)^T\Pi_{n+1} + D_t\Pi\,(x_n,u_n) \;=\; 0,
\end{aligned} \tag{35}$$

$$u_{n+1} = -R^{-1}B^T(x_n, u_n)\Pi_{n+1}x_n - \sum_{i=1}^{k}(u_n)_i\left(\frac{\partial B_{1\to m,i}}{\partial u}(x_n, u_n)\right)^T \Pi_{n+1}x_n \tag{36}$$

(or an iterative form similar to equations (33-34) but including the extra derivative terms). One simple way to obtain a numerical approximation of the time derivative of Π in this equation is with the difference formula

$$D_t\Pi\left(x_n, u_n\right) = \left(\Pi_{n+1} - \Pi_n\right) / \left(t_{n+1} - t_n\right). \tag{37}$$

Since equations (35-36) are solved online, the previous value of Π will be available so that this calculation can be done, while with the offline power series calculations there is no way to evaluate $D_t\Pi$ and so it must be assumed small and ignored. Equation (35) obviously cannot be solved with an algebraic Riccati equation solution algorithm, but must be solved using some more general zero-finding technique for nonlinear systems (such as a Newton-Raphson algorithm).

As an further control variation, if we are using a general zero-finding algorithm, we can choose to apply it to the combined system,

$$\begin{aligned}
&\Pi_{n+1}A(x_n) + A^T(x_n)\Pi_{n+1} - \Pi_{n+1}B(x_n, u_{n+1})R^{-1}B^T(x_n, u_{n+1})\Pi_{n+1} + Q \\
&+ \sum_{i=1}^{m}(x_n)_i\left(\frac{\partial A_{1\to m,i}}{\partial x}(x_n)\right)^T \Pi_{n+1} + \sum_{i=1}^{k}(u_{n+1})_i\left(\frac{\partial B_{1\to m,i}}{\partial x}(x_n, u_{n+1})\right)^T \Pi_{n+1} \\
&- \Pi_{n+1}B(x_n, u_{n+1})R^{-1}\sum_{i=1}^{k}(u_{n+1})_i\left(\frac{\partial B_{1\to m,i}}{\partial u}(x_n, u_{n+1})\right)^T \Pi_{n+1} + D_t\Pi\left(x_n, u_{n+1}\right) = 0,
\end{aligned} \tag{38}$$

$$u_{n+1} + R^{-1}B^T(x_n, u_{n+1})\Pi_{n+1}x_n + \sum_{i=1}^{k}(u_{n+1})_i\left(\frac{\partial B_{1\to m,i}}{\partial u}(x_n, u_{n+1})\right)^T \Pi_{n+1}x_n = 0, \tag{39}$$

finding both the Π and u updates simultaneously instead of first finding Π in (35) and then updating u in (36).

An approach using cheap control, as described in equations (29-30), is also an option with control through online gain matrix evaluations. The idea again is that restructuring the problem may decrease the effects of potential pitfalls due to the control nonlinearities, while hopefully not altering the problem so much that the resulting control is ineffective for the original system. This may be particularly useful when dealing with a hard-to-control scenario such as the one described below.

One aspect of nonlinear-u dynamical systems which presents an obstacle to feedback control implementation is related to the fact that there may be values of u which result in the control-dependent B matrix becoming zero. The system becomes uncontrollable at the specific point where $B(x) = 0$, and the control formula in equation (21) is reduced to $u(x) = 0$. However, this situation can have effects on some of the formulas we are studying even away from the zero point itself. Examples One and Two in the sections to follow are problems which contain a zero-valued B for certain u values. This results in a discontinuity in the u update iteration generated by the online-update SDRE algorithm in equations (33-34). Depending on the specifics of the example involved, this may cause major problems in obtaining convergence of that algorithm (and the related algorithms in (35-36) and (38-39) which involve iterative processes as well) in some region around the zero point. This difficulty will be illustrated on Example One in the next section.

The effectiveness of the power series method also may be impaired by an uncontrollable $B(u) = 0$ point if the iterative form of the method in equation (28) is used, since its convergence may become uncertain. In contrast, other control techniques may not be affected by this behavior; the polynomial form of the power series method and also the cheap control version of the method should be more likely to succeed in spite this obstacle, as they are based on direct evaluation of equations instead of an iterative process, (as well as finding their control values through an approximation rather than the original system itself). A cheap control version of the online-evaluation method will be working with a modified version of the system, but will still be using an iterative evaluation process which may have difficulties in converging.

In sum we have a number of different algorithms which are variations on the general method using the state-dependent Riccati equation. Those in this section are based on the re-evaluation of the SDRE over and over during the control process, while those in the previous section are based on using a power series approximation of the SDRE to simplify the control evaluation so that it can be done primarily offline. Some of these variations may be more effective than others at controlling systems with both state and control nonlinearities, and overcoming difficulties created by situations like the zero point discussed above. This will be studied in the example simulations in the following sections.

5 EXAMPLE PROBLEM ONE

The control algorithms discussed here were implemented in Matlab programs on a set of example problems to test their ability to control nonlinear-u systems. The first of these examples is a modified version of a model of the flight dynamics of a high-performance aircraft, taken from Garrard, Enns and Snell [15]. The model contains five state variables representing the flight conditions of the aircraft: x_1 is the deviation of the velocity from its trim value (given in units of (100m)/s), x_2 is the deviation of the angle of attack from trim (in radians), x_3 is the pitch rate (rad/s), x_4 is the flight path angle (radians), and x_5 is the deviation of the canard (control flap) deflection angle from trim (radians). The single control u is the input canard deflection in radians.

The model is quadratically nonlinear in the state variables, and we have added a (non-physical) quadratic nonlinearity (involving B_2 as seen below) to the control dynamics to test the ability of the algorithms to stabilize a system of this type. The system is given by

$$\dot{x} = (A_0 + x_2 A_{NL})\, x + (B_1 + uB_2)\, u, \tag{40}$$

where the matrices A_0, A_{NL}, B_1 and B_2 are all constant-valued, and are given by:

$$A_0 = \begin{bmatrix} -0.0443 & 1.1280 & 0.0 & -0.0981 & 0.0 \\ -0.0490 & -2.5390 & 1.0 & 0.0 & -0.0854 \\ -0.0730 & 19.3200 & -2.2700 & 0.0 & 22.6834 \\ 0.0490 & 2.5390 & 0.0 & 0.0 & 0.0854 \\ 0.0 & 0.0 & 0.0 & 0.0 & 20.0 \end{bmatrix},$$

$$A_{NL} = \begin{bmatrix} -0.2317 & 0.0 & 0.0 & 0.0 & 0.0 \\ -1.2760 & -0.7922 & 0.0 & 0.0 & 0.0206 \\ 0.1020 & 64.2940 & -13.9710 & 0.0 & -5.4167 \\ 1.2760 & 0.7922 & 0.0 & 0.0 & -0.0206 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix},$$

$$
\begin{aligned}
B_1 &= \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 20.0 \end{bmatrix}^T, \\
B_2 &= \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 2.0 \end{bmatrix}^T.
\end{aligned} \tag{41}
$$

The cost functional to be minimized (also modified from that used in [15]) is

$$
J(x_0, u) = \frac{1}{2} \int_0^\infty \left(x^T Q x + u^2 \right) dt, \tag{42}
$$

where the state weight matrix is

$$
Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix}. \tag{43}
$$

The SDRE control techniques were applied to this example with the initial state chosen as $x_0 = [0 \;\; 25(\pi/180) \;\; 0 \;\; 0 \;\; 0]^T$, a large angle of attack which must be brought back to trim, as efficiently as possible with respect to J. The main variations of the SDRE tested were "cheap control" and standard formulations of the dynamic system, and power series approximation (with varying number of terms) and online-evaluation approaches to solving the SDRE. The power series method was applied using one, two, and three power series terms. Also, the non-cheap power series method was implemented with u found both through an iterative solution of equation (28) for u, and a polynomial evaluation of equation (27), as described in Section 3. In the polynomial evaluation, the choice of root was that with the smallest absolute value.

As mentioned in Section 4, some nonlinear-u systems have the property that $B(u) = 0$ at certain nonzero values of u, rendering the system uncontrollable at those points and creating the potential for problems in some of the algorithms. This feature is illustrated for the aircraft dynamics model in two figures showing the control update mapping $u_{n+1} = F(u_n)$ representing the evaluation of equations (31-32) – or a single pass through the iterative scheme (33-34). Figure 1 evaluates this mapping at state $x = [0 \;\; 25(\pi/180) \;\; 0 \;\; 0 \;\; 0]^T$, with the discontinuity present at $u_n = -10$ (the point at which $B = 0$). At this 25° angle of attack the discontinuity still allows a solution to the update equations (33-34) – that is, a control value where the iterative process has converged so that $F(u_n) = u_n$. (In fact there are two possible solutions, which is an interesting problem but still allows for the system to be controlled.) However, if the same update mapping is examined at a 35° angle of attack, the influence of the discontinuity is broader and there is no solution to the equations, as Figure 2 shows. In that case the iterative algorithm in equations (33-34) will not be able to converge, and it will be nearly impossible for the algorithm to control the system successfully.

In contrast with the success of equations (33-34) at a 25° angle, attempts to use a Newton-Raphson algorithm in the full nonlinear online-update method (35-36) and the combined gain/control update method (38-39) were unable to converge and find a gain matrix solution. The inclusion of the more complex dynamics and/or the less specific numerical algorithm make the solution of this system of equations much more difficult. The approaches using the Newton-Raphson algorithm were discarded as a result of this in favor of the methods more

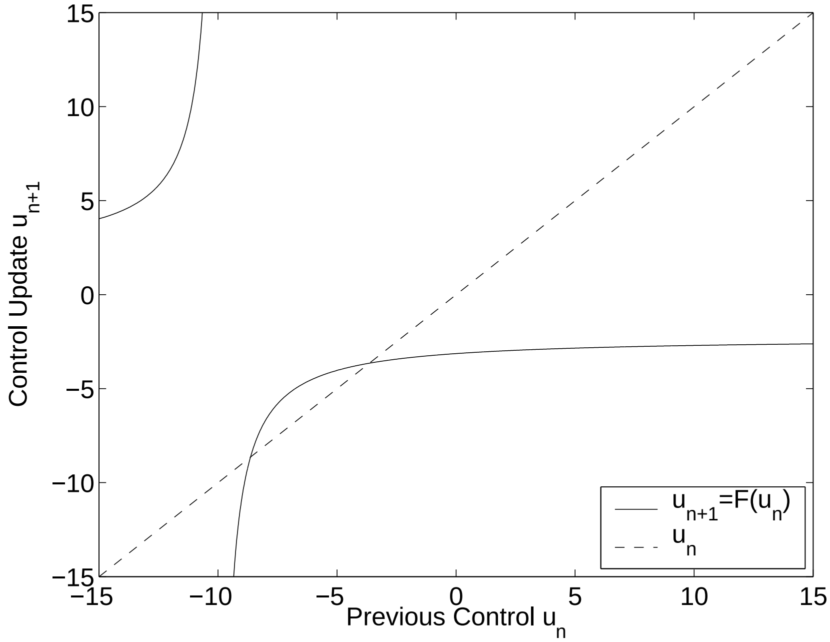

Figure 1: *Control update mapping, with identity mapping, at 25° angle of attack.*

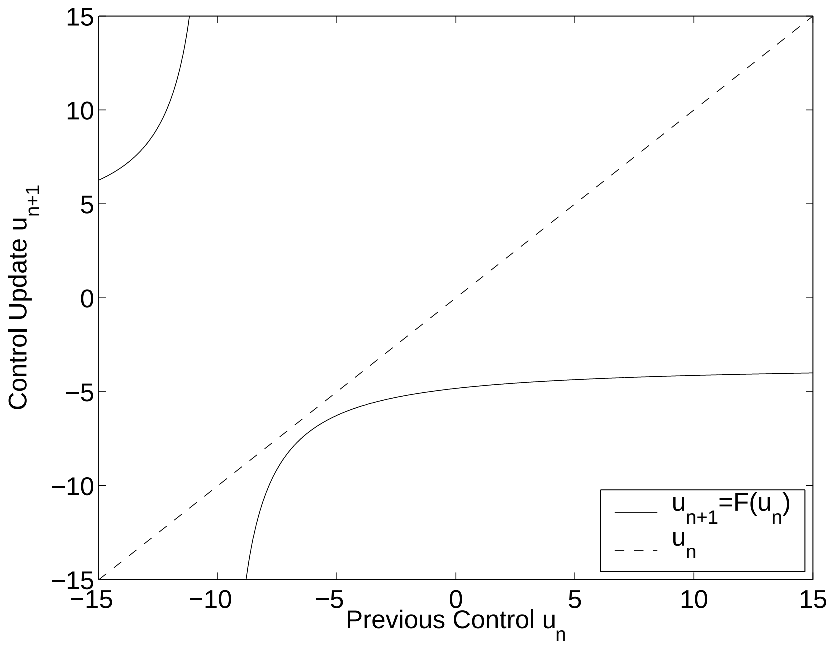

Figure 2: *Control update mapping, with identity mapping, at 35° angle of attack.*

closely related to the Riccati equation (and implemented using a built-in Matlab Riccati equation solver), which appear to be significantly more robust.

The cost values resulting from each of these methods are given in Table 1, as is the result of a linear Riccati control found by linearizing the system about $x = 0$, $u = 0$. The most

Table 1: *Cost functional values for SDRE control algorithms on Example One.*

Method	Cost	Cost with Noise
Cheap Power Series (1 term)	5.448	5.672
Cheap Power Series (2 term)	5.578	5.809
Cheap Power Series (3 term)	5.580	5.797
Plain Power Series (1 term, iterative)	4.426	4.547
Plain Power Series (2 term, iterative)	4.470	4.580
Plain Power Series (3 term, iterative)	4.479	4.562
Plain Power Series (1 term, polynomial)	4.426	4.540
Plain Power Series (2 term, polynomial)	4.469	4.615
Plain Power Series (3 term, polynomial)	4.479	4.588
Cheap Online Evaluation	4.460	4.605
Plain Online Evaluation	4.585	4.675
Linearized Control	4.641	4.807

efficient method is the standard form power series control, with the cheap online-evaluation control having a cost only very slightly higher than that. The standard form version of the online-evaluation control is also fairly close, while the cheap power series control is significantly off from the other three variations and the linear control. All show a clear ability to stabilize the aircraft from an initial 25° angle of attack. Changing the number of terms used in the power series methods seems to have only a small effect in this case, and the polynomial and iterative solutions for the standard form power series algorithm prove to be essentially identical here.

In contrast with the power series method, in the online-evaluation control the alteration to use cheap control increases the effectiveness of the algorithm on this example problem. The reason may be that the controllability issues are minimized through this modification of the system. In contrast, the power series method was not expected to be affected as much by the controllability problem. Thus the addition of cheap control would not be expected to help much, and it may have detracted from the method's effectiveness due to it finding u values based on a system slightly different from the original.

For this example a good choice of weight in the cheap control cost functional (30) was $\tilde{R} = 10^{-4}$. Other values were less effective, as larger ones apparently moved the cheap control system too far from the original problem, while smaller ones apparently created problems with $\tilde{R}^{-1}$ blowing up.

The norm of the state variables, $x^T Q x$, is plotted in Figure 3 for the four main versions of the SDRE algorithm. The most efficient of the versions with different numbers of power series terms and calculation methods are plotted: cheap control with one term, and standard control with one term and the polynomial solution. The two online-evaluation control results

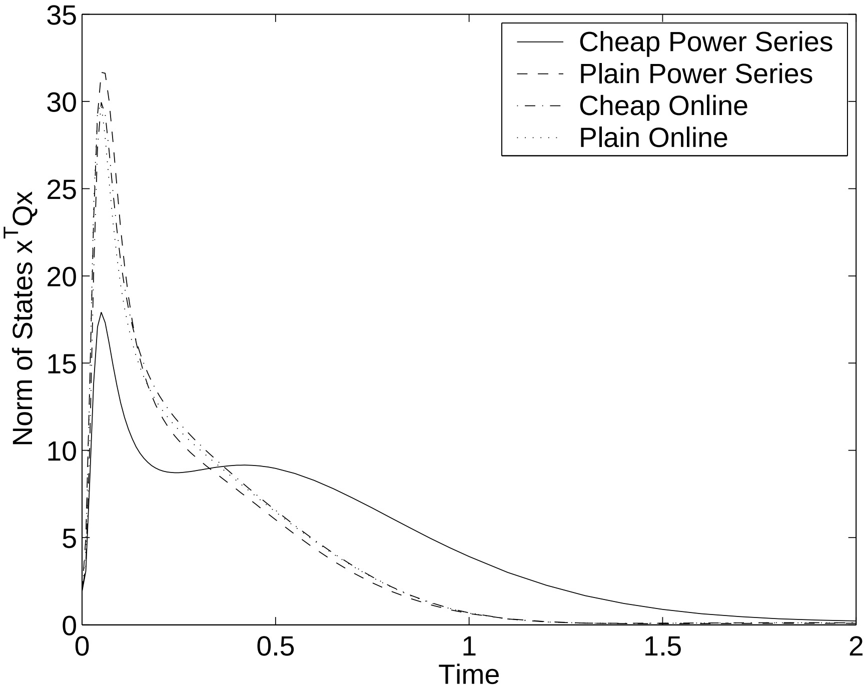

Figure 3: *Comparison of state norms in Example One with several control methods.*

are shown as well. The control inputs calculated by these same methods are plotted in Figure 4. The values of the control u remain greater than the problematic value of -10 where the system is uncontrollable. Note that these figures (as well as later ones) display only the first few seconds of the state and control trajectories. The simulations themselves were carried out through an interval of 10 seconds to allow for more complete convergence, but since the important dynamics occur in the early time interval, that is the region shown for clarity.

The various control methods were also applied to the same model with a uniform random noise source $|\delta(t)| < 2.5(\pi/180)$, ten percent of the initial nonzero state, added to the dynamics to see how the methods would compensate for this in controlling the problem. In this example the results are very similar to those without noise, with the methods' costs falling in the same order and the plain form of the power series method having the lowest cost overall. In each case the cost has increased by a noticeable amount but the system is stabilized without too much difficulty, as the divergence due to noise in this case seems to have made the problem only slightly harder to control. These costs are listed in Table 1 along with the noise-free case.

Simulations using one selection from each class of control algorithms were run with larger initial angles of attack, to see which of the methods would prove effective at stabilizing the system over a larger range of conditions. The three one-term power series methods were used, as were the two online-evaluation methods. The same process was followed as described above, with initial angles of 35°, 45°, and 55°. The resulting cost functional values are given in Table 2.

The various algorithms drop out one by one as the initial angle becomes larger and the

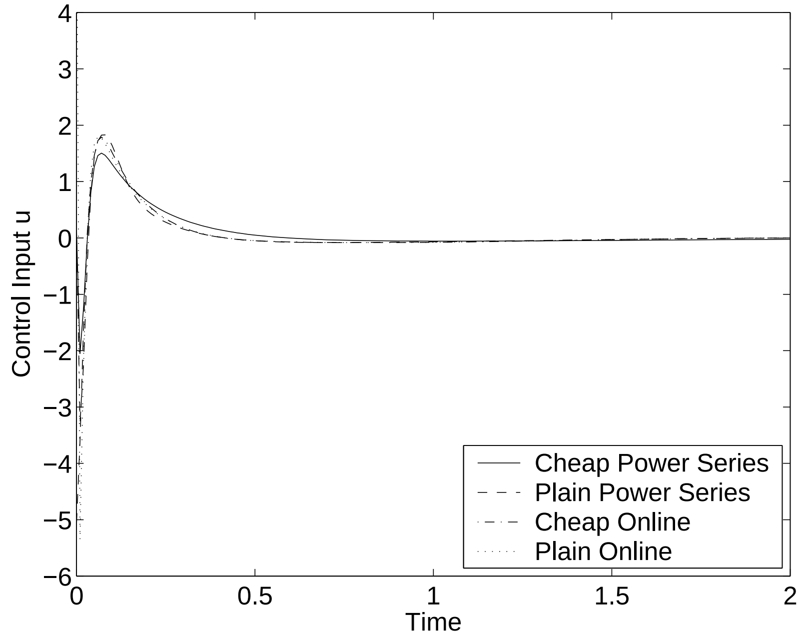

Figure 4: *Comparison of control inputs for Example One with several control methods.*

Table 2: *Cost functional values with larger initial angles in Example One.*

Method	Cost
Initial Angle 35°	
Cheap Power Series (1 term)	12.064
Plain Power Series (1 term, iterative)	(fails)
Plain Power Series (1 term, polynomial)	10.678
Cheap Online Evaluation	11.160
Plain Online Evaluation	(fails)
Linearized Control	11.370
Initial Angle 45°	
Cheap Power Series (1 term)	22.644
Plain Power Series (1 term, polynomial)	25.036
Cheap Online Evaluation	(fails)
Linearized Control	(fails)
Initial Angle 55°	
Cheap Power Series (1 term)	38.550
Plain Power Series (1 term, polynomial)	(fails)

problem becomes more difficult to control. More than a mere increase in the magnitude of the state value to be stabilized, there is something of a qualitative change in the scenario. As noted above, at a 35° angle of attack the online-evaluation iterative control update becomes

unsolvable (while at 25° it has a findable solution). The other algorithms may also be affected to some degree by the element of uncontrollability in the problem as the range of angles considered expands. The control values involved in some algorithms may approach the danger point of -10 and as a result fail to converge to an effective control.

The two standard power series approaches had been the most effective at stabilizing the system from a 25° angle and had virtually identical results in that case. In these additional simulations, as the system gets more difficult to control, the iterative version of the power series fails at 35°, but the polynomial version is adaptable to a much larger range. It is still successful at 45°, though the cheap power series control has overtaken it in terms of cost. Eventually, at 55° only the cheap power series algorithm can successfully control the system. Thus the least efficient of the methods at small angles is the one able to stabilize this example over the widest range of conditions using its slow-but-steady type of control. This can be seen in Figure 3 and also in Figure 5, which plots the state norms for the higher 35° angle. Also note that as the angle increases, so does the size of the initial spike of

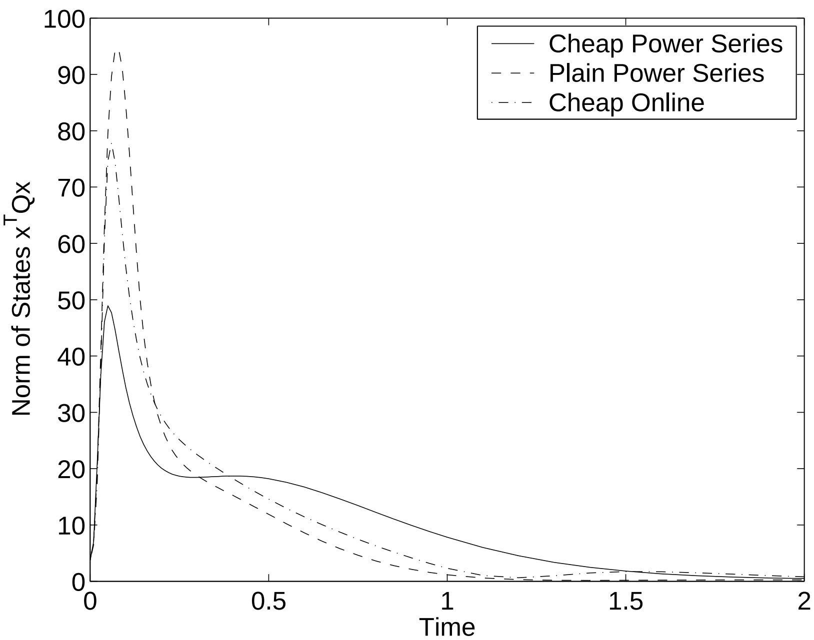

Figure 5: *State norms in Example One for a 35° initial angle.*

activity, as the algorithms seek to bring the angle down quickly. This spike is much greater for the other methods than for the cheap power series, making it unsurprising that they would become unstable sooner at high angles while the cheap power series method is able to continue operating in that range.

6 EXAMPLE PROBLEM TWO

The second example problem used to test the control algorithms was taken from Wernli and Cook [4], in which the use of power series approximations was proposed for finding a gain matrix from the state-dependent Riccati equation. That approach was applied to the

following system, and here we use our versions of the SDRE method on it. This is a small two-state system with one control:

$$\begin{aligned} \dot{x}_1 &= x_2 && (44) \\ \dot{x}_2 &= x_1 x_2^2 + \left(1 + \sqrt{|x_1|}\right)\left(2u + u^2\right), && (45) \end{aligned}$$

which the control methods will try to stabilize in such a way that the cost functional

$$J = \frac{1}{2}\int_0^\infty \left(2x_1^2 + x_2^2 + u^2\right) dt \tag{46}$$

is minimized.

The quasi-linearization chosen by Wernli and Cook to set up the SDRE form is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ x_2^2 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \left(1 + \sqrt{x_1}\right)(2 + u) \end{bmatrix} u. \tag{47}$$

Using this form of the system the various SDRE methods described in this paper are implemented in an attempt to control the system from three initial states: $x_0 = [0.5 \ \ 0.0]^T$, $x_0 = [1.0 \ \ 0.0]^T$, and $x_0 = [0.5 \ \ 0.5]^T$, the three states used by Wernli and Cook in their paper. As with Example One, there is a danger of possible uncontrollability hindering stabilization of the system, in this case near the point $u = -2$ where $B(u) = [0 \ \ 0]^T$.

The cost functional results are presented in Table 3 for the three chosen initial conditions and the various control techniques. The best values in [4] (found using a standard-form

Table 3: *Cost functional values for SDRE control algorithms on Example Two.*

Method	Initial Conditions			
	[0.5 0.0]	[1.0 0.0]	[0.5 0.5]	[0.5 0.5] w/Noise
Cheap Power Series (1 term)	0.2593	1.0322	0.4076	0.4114
Cheap Power Series (2 term)	0.2724	1.0891	0.4511	0.4566
Cheap Power Series (3 term)	0.2714	1.1115	0.4236	0.4289
Plain Power Series (1 term, iter)	0.2669	(fails)	0.6671	0.6604
Plain Power Series (2 term, iter)	(fails)	(fails)	(fails)	(fails)
Plain Power Series (3 term, iter)	0.2591	(fails)	0.7566	0.7455
Plain Power Series (1 term, poly)	0.2669	(fails)	(fails)	(fails)
Plain Power Series (2 term, poly)	0.2594	(fails)	0.5339	0.5363
Plain Power Series (3 term, poly)	(fails)	(fails)	(fails)	(fails)
Cheap Online Evaluation	(fails)	(fails)	(fails)	(fails)
Plain Online Evaluation	0.2673	1.1905	0.9214	(fails)
Linearized Control	0.2580	1.1297	0.7069	0.7094

power series method) are: for $x_0 = [0.5 \ \ 0.0]^T$, $J = 0.256$; for $x_0 = [1.0 \ \ 0.0]^T$, $J = 1.11$; for $x_0 = [0.5 \ \ 0.5]^T$, $J = 0.604$. In this example the cheap power series form of the SDRE method is most efficient at controlling the system. It results in cost values which are almost

the same as (for $[0.5 \;\; 0.0]^T$) or less than (for the other two conditions) the values found in [4] or with any of the other algorithms here.

The controls generally have a more difficult time in stabilizing this example than they did with Example One, as some methods fail to control the system, especially at the larger initial condition $[1.0 \;\; 0.0]^T$. The cheap online-evaluation control does not succeed with any of the three initial states, and the standard power series controls only succeed intermittently for various states and numbers of series terms. The standard online-evaluation control does stabilize all three cases. However, though it is fairly close to the cost of the cheap power series control for initial conditions of $[0.5 \;\; 0.0]^T$ and $[1.0 \;\; 0.0]^T$, it is much larger for $[0.5 \;\; 0.5]^T$. Control of this nonzero-x_2 condition seems to be more complex, as the cost functional values are much more widely spread in comparison to the smaller variation between methods with the other two initial conditions. This can also be seen in the linearized-control results, which are very good for $[0.5 \;\; 0.0]^T$ and $[1.0 \;\; 0.0]^T$ but fall off for the more complex case of $[0.5 \;\; 0.5]^T$. In the cheap power series control, the one-term version is the most efficient.

The state variables for selected methods are plotted in Figure 6 for initial state $[0.5 \;\; 0.5]^T$. The algorithms plotted are the cheap power series control with one term, the standard power

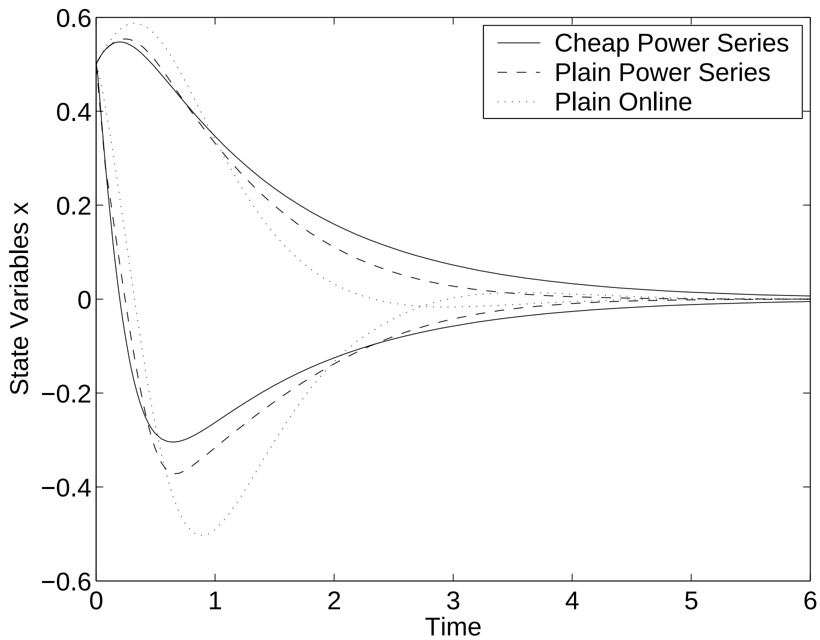

Figure 6: *Comparison of state variables in Example Two with several control methods.*

series control with two terms (and polynomial solution), and the standard online-evaluation control. The control inputs for these same methods are presented in Figure 7. It can be seen that each of these successful controls stay away from the $u = -2$ value where the system becomes uncontrollable (though the standard online-evaluation control comes very close). The methods which fail to stabilize the system may be encountering difficulty with that aspect of the problem when trying to exert a more powerful control influence.

A set of simulations was done with initial condition $[0.5 \;\; 0.5]^T$, and the addition of uniform noise $|\delta(t)| < 0.05$. The results, given in Table 3, align closely with those of that

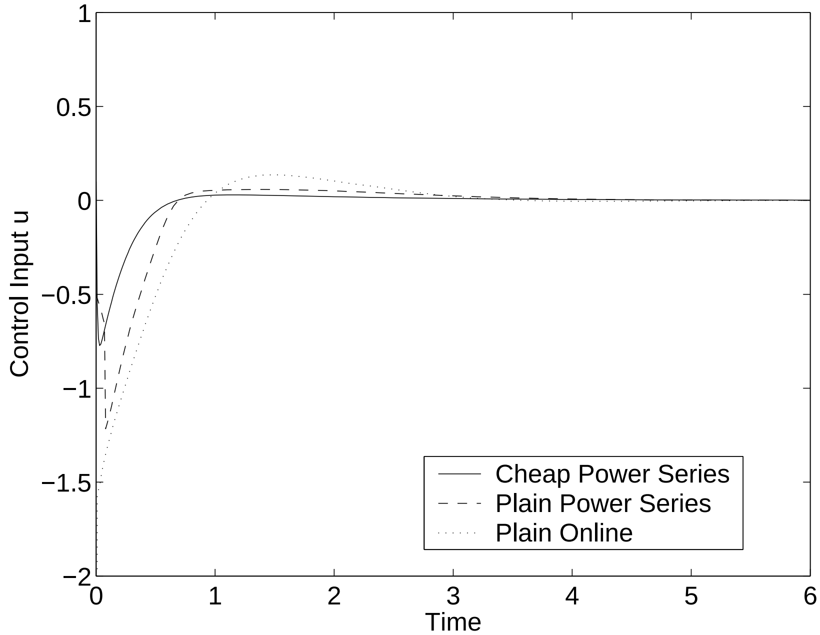

Figure 7: *Comparison of control inputs for Example Two with several control methods.*

scenario without noise, with the exception that the standard online-evaluation control fails in the case with noise. The cheap power series method (using one term) is still the most efficient.

7 EXAMPLE PROBLEM THREE

A third example, taken from a paper by Lin [16], describes an angular momentum equation for a rigid body with non-affine control inputs. The system contains three states and two controls, including both linear and highly nonlinear control input terms:

$$\begin{aligned} \dot{x}_1 &= -x_2 x_3 + u_1 &(48)\\ \dot{x}_2 &= 2x_1 x_2 x_3 + u_2 &(49)\\ \dot{x}_3 &= x_1 x_2 + x_1 x_2^3 \left(1 - cos x_3\right) \left(e^{(u_1 u_2)^2} - 1\right). &(50) \end{aligned}$$

Lin's control approach is stability-based rather than performance-based using a cost functional, so for our purposes we introduce one, given by

$$J = \frac{1}{2} \int_0^\infty \left(x_1^2 + x_2^2 + 10x_3^2 + u_1^2 + u_2^2\right) dt. \tag{51}$$

In quasi-linearizing this example, there is no controllability issue as in the first two examples where $B(u) = 0$ at some nonzero u. However, the control term in the $\dot{x}_3$ equation is notably troublesome due to its extreme nonlinearity. Control inputs with magnitude greater than 1 can have very large effects on the system dynamics. We write the system in

the desired SDRE form as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & -x_3/2 & -x_2/2 \\ 2x_2x_3/3 & 2x_1x_3/3 & 2x_1x_2/3 \\ x_2/2 & x_1/2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + B \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{52}$$

The choice of B is a difficult one, since the $\dot{x}_3$ control term is not a pure polynomial which could be easily split up. One approach is to modify this term by multiplying and dividing it by u_1 to obtain

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x_1x_2^3(1 - cosx_3)\left(e^{(u_1u_2)^2} - 1\right)/u_1 & 0 \end{bmatrix}, \tag{53}$$

but this has an inherent problem in calculating B when u_1 approaches 0. Other versions which were tried include a similar formulation in terms of u_2, a formulation which uses both – splitting that term from (50) into two terms similarly to the way the state nonlinearities are split in (52) – and a formulation which switches between the two forms, keeping the u_i value farthest from zero as the one by which the original term is divided.

However, none of these remove the difficulty entirely, but merely superficially modify it to try to lessen its impact. We can also alter the problem more significantly, creating an approximation to the actual formula for use in finding the control, and then implementing that control back on the original system. With this example this was done by writing the exponential term as a power series,

$$e^{(u_1u_2)^2} = 1 + (u_1u_2)^2 + \frac{1}{2}(u_1u_2)^4 + \frac{1}{6}(u_1u_2)^6 ... \tag{54}$$

and using only the first two or three terms to approximate it. Substituting the truncated series into the system results in a two-term approximation of

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x_1x_2^3(1 - cosx_3)u_1u_2^2/2 & x_1x_2^3(1 - cosx_3)u_1^2u_2/2 \end{bmatrix}, \tag{55}$$

or a three-term approximation of

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x_1x_2^3(1 - cosx_3)(u_1u_2^2 + u_1^3u_2^4/2)/2 & x_1x_2^3(1 - cosx_3)(u_1^2u_2 + u_1^4u_2^3/2)/2 \end{bmatrix}. \tag{56}$$

(In these formulations the control term has been split into multiple parts, as the state dynamics terms have been in equation (52), to spread the effects further over the A and B matrices.) This alteration of the problem creates a different approach which might be more stable than the volatile original form.

Simulations were run using an initial state of $x = [1\ \ \text{-}1\ \ \text{-}1]^T$, with each of the control algorithm variations. Due to the type of control nonlinearity here having different implementation and controllability issues, as described above, the results were quite different from the previous examples. All of the non-cheap control algorithms fail to stabilize the system,

while the cheap control methods have more success, probably due to the movement of the exponential from the control to the state dynamics reducing its troubling effects. Trying to control the system using the exponential nonlinearity as is proved to be mostly ineffective, while the use of a power series approximation for that nonlinearity in finding the control improved things significantly. The cost functional values for the successful methods are given in Table 4, also noting the specific form of exponential used in each method. There

Table 4: *Cost functional values for SDRE control algorithms on Example Three.*

Method	Cost	Cost with Noise
Cheap Power Series (1 term, u_2 form)	32.33	(fails)
Cheap Power Series (2 terms, u_2 form)	42.06	(fails)
Cheap Power Series (3 terms, u_2 form)	61.17	(fails)
Cheap Power Series (1 term, 1-term form)	32.88	32.44
Cheap Power Series (2 terms, 1-term form)	39.79	48.85
Cheap Power Series (3 terms, 1-term form)	29.33	31.03
Cheap Power Series (1 term, 2-term form)	34.47	34.19
Cheap Power Series (2 terms, 2-term form)	28.15	(fails)
Cheap Power Series (3 terms, 2-term form)	28.37	29.31
Cheap Online Evaluation (1-term form)	8.87	8.80
Cheap Online Evaluation (2-term form)	6.91	8.77

are no linearized-control results in this example because the extremely nonlinear nature of it makes impossible the finding of a viable control through linearization as in the previous sections.

The following versions of the cheap power series method have the norm of their resulting state vectors plotted in Figure 8: that using the u_2 form of the exponential term (and one term in the SDRE equation power series expansion), the one-term exponential expansion (with three SDRE series terms), and the two-term exponential expansion (with two SDRE series terms). Also plotted are the cheap online-evaluation methods with one and two exponential expansion terms. The control inputs for these methods are shown in Figure 9.

Although there is some variation between the different implementations of the cheap power series method, the cheap online-evaluation methods produce dramatically more efficient results that any of them. The power series algorithms seem to have a pause of sorts in the control, as the way those particular methods interact with this example's dynamics seemingly leads to the control influence being applied gradually, with some of the major effects not appearing until three or four seconds into the simulation. In contrast, the online-evaluation controls have a bigger initial burst of control influence, which leads to the system being largely stabilized after only one or two seconds. Of the different approaches to treating the exponential control nonlinearity, the two-term power series approximation was the most effective for both the SDRE power series and online-evaluation algorithms. Strangely the best choice for number of terms in the power series representing the SDRE solution varied, with a different number producing the lowest cost for each different way of dealing with the exponential. The success of the online-evaluation method may be partly a consequence of

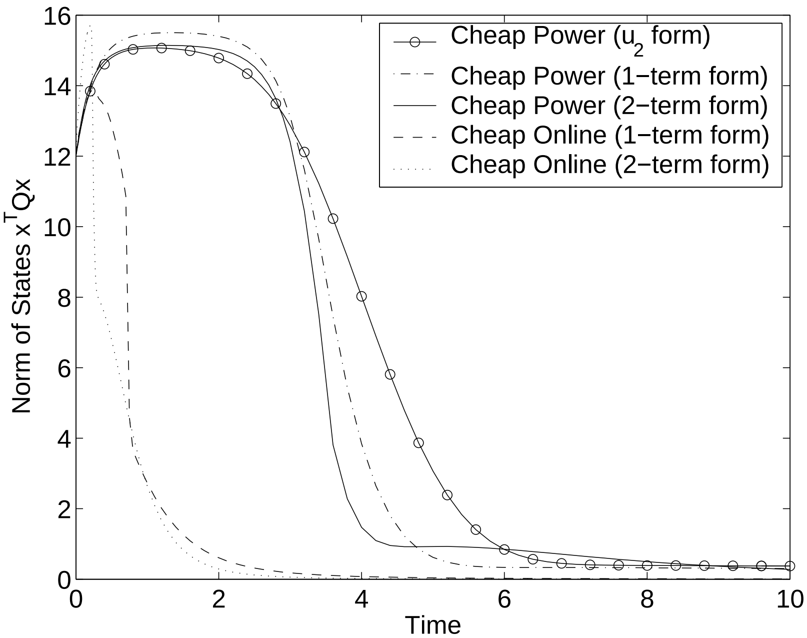

Figure 8: *Comparison of state variables in Example Three with several control methods.*

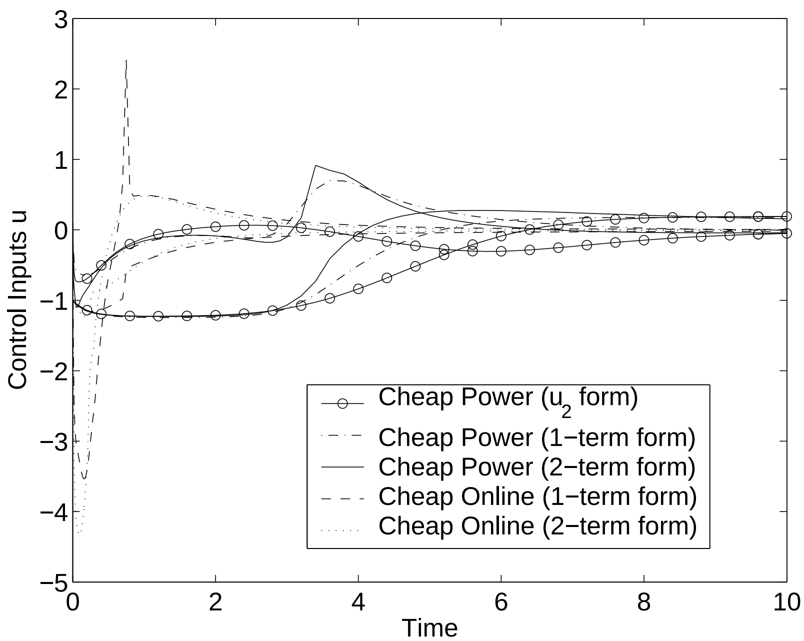

Figure 9: *Comparison of control inputs for Example Three with several control methods.*

this example not containing a controllability danger as the previous two did, making it easier for that version of the SDRE method to construct a strong control using large initial control input values.

As in the previous two examples, these simulations are repeated with an added uniform noise, $|\delta(t)| < 0.1$, in the system dynamics. The cost values for these simulations are given in Table 4 as well. The online-evaluation controls again perform much better than the power series controls. There are changes among the different methods, though; the u_2 form of the exponential now fails to produce a viable control, and the cheap power series controls with two terms representing the SDRE both are much less effective than they had been without the noise. Though the form of the online-evaluation control with two terms for the exponential is still the most effective, it also is reduced somewhat in comparison with the one-term version which handles the noise easily.

8 CONCLUSIONS

The test problems studied, each nonlinear in both the state and control, were each successfully stabilized by versions of the state-dependent Riccati equation based feedback control method. The power series method, with a cheap control formulation moving the nonlinearities all into the state dynamics, was the most consistent method overall, successfully controlling all three test problems. The cheap control version of the online SDRE evaluation method was much more efficient than the others at controlling Example Three, but was vulnerable to failure with the other two examples.

In light of these results no one specific method appears to be an obvious best choice. For problems which may have difficulties with controllability at certain points, these simulations suggest that the more stable cheap power series version of the SDRE method is probably best, while for problems without such a concern, the cheap online-evaluation version might be more efficient. The forming of a cheap control system to move any control nonlinearities into the state dynamics seems to be a quite helpful tool in preparing the system for SDRE regulation. The different SDRE control variations should be kept in mind when approaching the control of a nonlinear system of the type studied here. Additional options are also contained in these methods, such as the number of terms to use in the power series approximating the SDRE gain matrix, or the way to write a certain example in the quasi-linearized SDRE form (including whether to make alterations to the system as was done in Example Three). These options can be used as seems appropriate for specific applications, as their effects will vary from system to system.

Despite these unanswered questions, the SDRE-based methods performed well. Future research plans in this area include the application of this control technique to higher-dimensional systems than those discussed in this paper, as well as possible work on using an SDRE-based method in combination with other control methods, similar to that in [17]. The goal for that investigation would be to combine methods in such a way that the speed and efficiency of the optimality-theory-based SDRE algorithms are largely retained while the stability and robustness properties of a different algorithm are merged into it as well, so that the strong control performance qualities noted in the examples above can be extended with confidence over a much broader domain.

REFERENCES

[1] Pearson, J. D., "Approximation Methods in Optimal Control," Journal of Electronics and Control, Vol 13, pp. 453-465 (1962).

[2] Garrard, W. L., McClamroch, N. H., and Clark, L. G., "An Approach to Sub-Optimal Feedback Control of Non-linear Systems," International Journal of Control, Vol 5, pp. 425-435 (1967).

[3] Burghart, J. A., "A Technique for Suboptimal Control of Nonlinear Systems," IEEE Transactions on Automatic Control, Vol 14, pp. 530-533 (1969).

[4] Wernli, A. and Cook, G., "Suboptimal Control for the Nonlinear Quadratic Regulator Problem," Automatica, Vol 11, pp. 75-84 (1975).

[5] Krikelis, N. J. and Kiriakidis, K. I., "Optimal Feedback Control of Non-linear Systems," International Journal of Systems Science, Vol 23, pp. 2141-2153 (1992).

[6] Cloutier, J. R., D'Souza, C. N., and Mracek, C. P., "Nonlinear Regulation and Nonlinear H_∞ Control Via the State-Dependent Riccati Equation Technique: Part 1. Theory," Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, FL, May 1996.

[7] Cloutier, J. R., D'Souza, C. N., and Mracek, C. P., "Nonlinear Regulation and Nonlinear H_∞ Control Via the State-Dependent Riccati Equation Technique: Part 2. Examples," Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, FL, May 1996.

[8] Mracek, C. P. and Cloutier, J. R., "Control Designs for the Nonlinear Benchmark Problem Via the State-Dependent Riccati Equation Method," International Journal of Robust and Nonlinear Control, Vol 8, pp. 401-433 (1998).

[9] Hammett, K. D., Hall, C. D., and Ridgely, D. B., "Controllability Issues in Nonlinear State-Dependent Riccati Equation Control," Journal of Guidance, Control and Dynamics, Vol 21, pp. 767-773 (1998).

[10] Beeler, S. C., Tran, H. T., and Banks, H. T., "Feedback Control Methodologies for Nonlinear Systems," Journal of Optimization Theory and Applications, Vol 107, pp. 1-33 (2000).

[11] Beeler, S. C., Tran, H. T., and Banks, H. T., "State Estimation and Tracking Control of Nonlinear Dynamical Systems," CRSC Technical Report CRSC-TR00-19, N.C. State University (2000).

[12] Cloutier, J. R., and Stansbery, D. T., "The Capabilities and Art of State-Dependent Riccati Equation-Based Design," Proceedings of the American Control Conference 2002, pp. 86-91.

[13] Anderson, B. D. O., and Moore, J. B., *Optimal Control: Linear Quadratic Methods*, Englewood Cliffs, New Jersey: Prentice-Hall, 1990.

[14] Lewis, F. L., and Syrmos, V. L., *Optimal Control*, New York: Wiley, 1995.

[15] Garrard, W. L., Enns, D. F., and Snell, S. A., "Nonlinear Feedback Control of Highly Manoeuvrable Aircraft," International Journal of Control, Vol 56, pp. 799-812 (1992).

[16] Lin, W., "Global Asymptotic Stabilization of General Nonlinear Systems with Stable Free Dynamics via Passivity and Bounded Feedback," Automatica, Vol 32, pp. 915-924 (1996).

[17] Sznaier, M., Cloutier, J., Hull, R., Jacques, D., and Mracek, C., "Receding Horizon Control Lyapunov Function Approach to Suboptimal Regulation of Nonlinear Systems," Journal of Guidance, Control, and Dynamics, Vol 23, pp. 399-405 (2000).

REPORT DOCUMENTATION PAGE				*Form Approved OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
01/07/2004	Contractor Report	

4. TITLE AND SUBTITLE	**5a. CONTRACT NUMBER**
State-Dependent Riccati Equation Regulation of Systems with State and Control Nonlinearities	
	5b. GRANT NUMBER NCC-1-02043
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Beeler, Scott C.	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER 23-762-65-AL

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
NASA Langley Research Center Hampton, VA 23681-2199 National Institute of Aerospace (NIA) 144 Research Drive Hampton, VA 23666	NIA Report No. 2004-08
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING/MONITOR'S ACRONYM(S)**
National Aeronautics and Space Administration Washington, DC 20546-0001	NASA
	11. SPONSORING/MONITORING REPORT NUMBER NASA/CR-2004-213245

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 64
Availability: NASA CASI (301) 621-0390 Distribution: Nonstandard

13. SUPPLEMENTARY NOTES

Langley Technical Monitor: David E. Cox

14. ABSTRACT

The state-dependent Riccati equations (SDRE) is the basis of a technique for suboptimal feedback control of a nonlinear quadratic regulator (NQR) problem. It is an extension of the Riccati equation used for feedback control of linear problems, with the addition of nonlinearities in the state dynamics of the system resulting in a state-dependent gain matrix as the solution of the equation. In this paper several variations on the SDRE-based method will be considered for the feedback control problem with control nonlinearities. The control nonlinearities may result in complications in the numerical implementation of the control, which the different versions of the SDRE method must try to overcome. The control methods will be applied to three test problems and their resulting performance analyzed.

15. SUBJECT TERMS

Feedback Control; Nonlinear Quadratic Regulator; State-Dependent Riccati Equation; Suboptimal Feedback Control

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	29	**19b. TELEPHONE NUMBER** *(Include area code)* (301) 621-0390